Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Today's topics:
  - **Comparison of EFA and CFA**
  - **CFA model parameters and identification**
  - CFA model estimation
  - CFA model fit evaluation

Confirmatory Factor Analysis (CFA)

- **The CFA unit of analysis is the ITEM (as in any LTMM):**

  $$\mathbf{y_{is}} = \boldsymbol{\mu}_\mathbf{i} + \boldsymbol{\lambda}_\mathbf{i}\mathbf{F_s} + \mathbf{e_{is}}$$ → **both items AND subjects matter**

  - Observed response for item *i* and subject *s*
    = intercept of item *i* (μ)
    + subject *s*'s latent trait/factor (F), item-weighted by loading λ
    + error (e) of item *i* and subject *s*

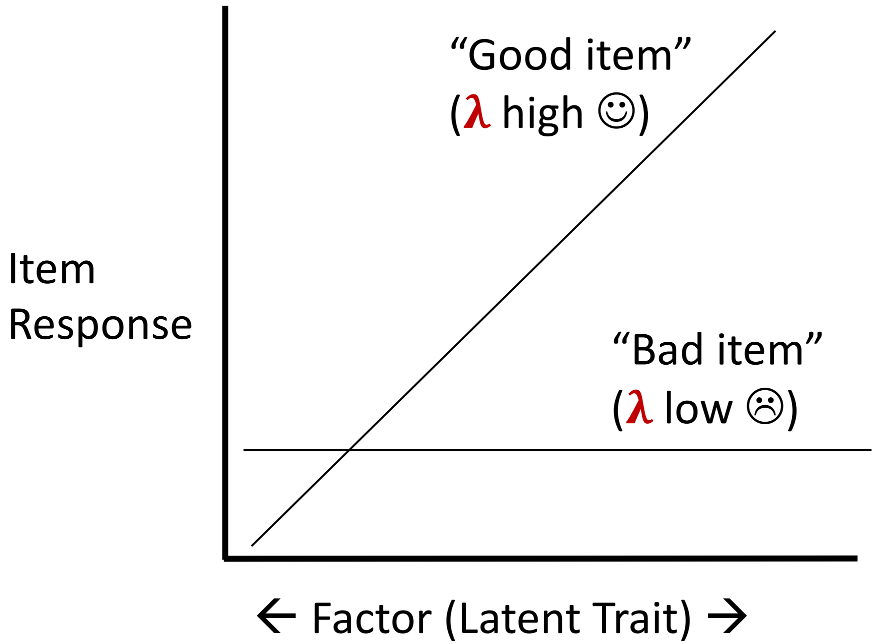
- **What does this look like? Linear regression (without a real X)!**

  - $y_s = \beta_0 + \beta_1 X_s + e_s$ → written for each item → $y_{is} = \beta_{0i} + \beta_{1i} X_s + e_{is}$
  - Intercept $\beta_{0i} = \mu_i =$
  - Slope of Factor $\beta_{1i} = \lambda_i =$
  - Residual $e_{is} = e_{is} =$

Why Item Intercepts Are Often Ignored…

A **"good" item** has a large slope (factor loading) in predicting the item response from the factor. Because this is a **linear slope**, the item is assumed to be equally discriminating (**equally good**) across the entire latent trait.

Similarly, a **"bad" item** has a flatter linear slope that is **equally bad** across the entire range of the latent trait.



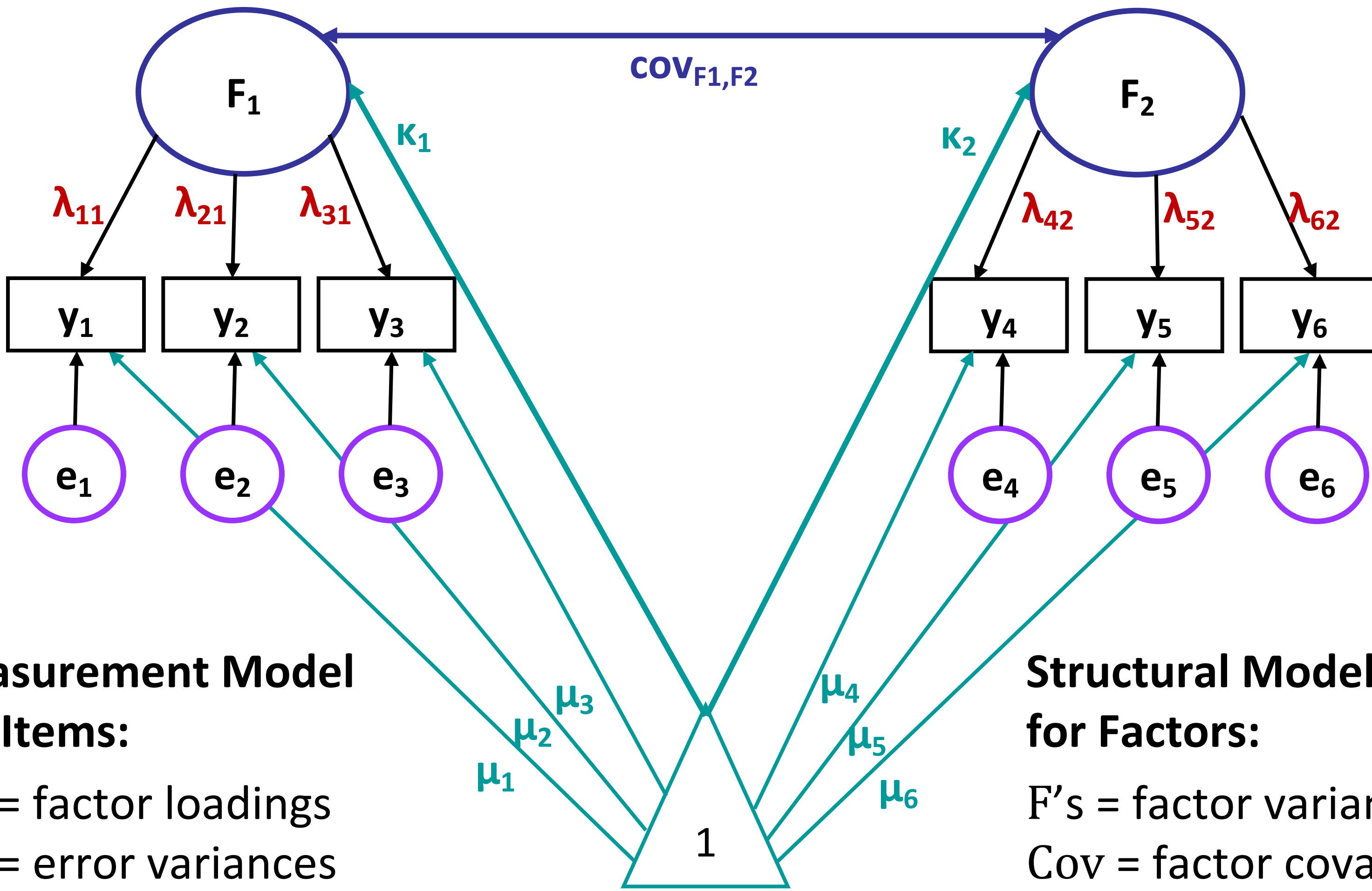
Here item intercepts are irrelevant in evaluating how "good" an item is, so they are not really needed.

But we will estimate them, because item intercepts are critical when:

- Testing factor mean differences in any latent factor model
- Items need to have a nonlinear slope in predicting the item response from the factor (IRT)

CFA Model with Factor Means and Item Intercepts

But some parameters will have to be fixed to known values for the model to be identified.



Measurement Model for Items:

λ's = factor loadings
e's = error variances
μ's = intercepts

Structural Model for Factors:

F's = factor variances
Cov = factor covariances
K's = factor means

2 Types of CFA Solutions

- <u>Unstandardized</u> → predicts scale-sensitive original item response:
  - **Regression Model:** $\mathbf{y_{is}} = \boldsymbol{\mu_i} + \boldsymbol{\lambda_i}\mathbf{F_s} + \mathbf{e_{is}}$
  - *Useful when comparing solutions across groups or time (when absolute values matter)*
  - Together, the model parameters predict the item means and ***item covariance matrix***
  - Note the solution asymmetry: item parameters $\boldsymbol{\mu_i}$ and $\boldsymbol{\lambda_i}$ will be given in the item metric, but $\mathbf{e_{is}}$ will be given as the error variance across persons for that item (squared metric)
  - $\mathbf{Var(y_i)} = [\boldsymbol{\lambda_i}^2 * \mathbf{Var(F)}] + \mathbf{Var(e_i)}$

- <u>Standardized</u> → Solution transformed to $\mathrm{Var}(\mathrm{y_i}) = 1, \mathrm{Var}(\mathrm{F}) = 1$ via :
  - *Useful when comparing items within a solution (relative values on same scale)*
  - Together, the standardized model parameters predict the **item correlation matrix**
  - Standardized intercept = $\boldsymbol{\mu_i}$ / $\mathbf{SD(y)}$ → not typically reported
  - Standardized factor loading = $[\boldsymbol{\lambda_i} * \mathbf{SD(F)}]$ / $\mathbf{SD(y)}$ **= item correlation with factor**
  - Standardized error variance = $\mathbf{1 - standardized}\ \boldsymbol{\lambda_i}^2$ = "variance due to *not* factor"
  - R^2 for item = $\mathbf{standardized}\ \boldsymbol{\lambda_i}^2$ = "variance due to the factor"

CFA Model Equations

- Measurement model per item (numbered) for subject *s*:
  - $y_{1s} = \mu_1 + \lambda_{11}F_{1s} + \mathbf{0}F_{2s} + e_{1s}$
  - $y_{2s} = \mu_2 + \lambda_{21}F_{1s} + \mathbf{0}F_{2s} + e_{2s}$
  - $y_{3s} = \mu_3 + \lambda_{31}F_{1s} + \mathbf{0}F_{2s} + e_{3s}$
  - $y_{4s} = \mu_4 + \mathbf{0}F_{1s} + \lambda_{42}F_{2s} + e_{4s}$
  - $y_{5s} = \mu_5 + \mathbf{0}F_{1s} + \lambda_{52}F_{2s} + e_{5s}$
  - $y_{6s} = \mu_6 + \mathbf{0}F_{1s} + \lambda_{62}F_{2s} + e_{6s}$

Here is the general matrix equation for these 6 item-specific equations:

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\lambda}\mathbf{F} + \mathbf{e}$$

where $\mathbf{Y}$, $\boldsymbol{\mu}$, and $\mathbf{e}$ = 6x1 matrices, $\boldsymbol{\lambda}$ = 6x2 matrix, and $\mathbf{F}$ =2x1 matrix

You decide **how many factors** and if each item has an estimated loading on each factor or not.

Unstandardized loadings ($\boldsymbol{\lambda}$) are the **linear slopes** predicting the item response (y) from the factor (F). **Thus, the model assumes a linear relationship between the factor and the item response.**

Standardized loadings are the slopes in a **correlation** metric (and Standardized Loading^2 = $\mathbf{R^2}$).

Intercepts ($\boldsymbol{\mu}$) are the expected item responses (y) when all factors = 0.

The Role of the CFA Model Parameters

- Data going in to be predicted by the CFA Model parameters
  = item covariance matrix (variances, covariances) and item means

- The CFA **item intercepts** (μ_i) predict the **item means**
  - Item means are unconditional; item intercepts are conditional on $F_s = 0$
  - When each item gets its own intercept (usual case), the item means will be perfectly predicted (so no room for mis-fit or mis-prediction)

- The CFA **item error variances** ($Var[e_{is}]$) predict the **item variances**
  - Item variances are unconditional; item error variances are conditional (leftover variance after accounting for the contribution of the factor)
  - When each item gets its own error variance (usual case), the item variances will be perfectly predicted (so no room for mis-fit or mis-prediction)

- The CFA **item factor loadings** (λ_i) predict the **item covariances**
  - Given 3+ items, there will be more covariances among items to predict than item factor loadings to predict them, **creating room for mis-fit**

CFA Model Predictions: (F_1 ~= y_1-y_3, F_2 ~= y_4-y_6)

Items from same factor (room for misfit or mis-prediction):

- Unstandardized solution: Covariance of y_1, y_3 = $\boldsymbol{\lambda_{11}} * \mathbf{Var(F1)} * \boldsymbol{\lambda_{31}}$
- Standardized solution: Correlation of y_1, y_3= $\boldsymbol{\lambda_{11}} * \mathbf{(1)} * \boldsymbol{\lambda_{31}}$ ← **std load**
- ONLY reason for correlation is their common factor (local independence, LI)

Items from different factors (room for misfit or mis-prediction):

- Unstandardized solution: Covariance of y_1, y_6 = $\boldsymbol{\lambda_{11}} * \mathbf{cov_{F1,\,F2}} * \boldsymbol{\lambda_{62}}$
- Standardized solution: Correlation of y_1, y_6= $\boldsymbol{\lambda_{11}} * \mathbf{cor_{F1,\,F2}} * \boldsymbol{\lambda_{62}}$ ← **std load**
- ONLY reason for correlation is the correlation between factors (again, LI)

Variances are additive (and will be reproduced correctly):

- **$Var(y_1) = (\lambda_{11}^2)*Var(F_1) + Var(e_i)$ → but note the imbalance of λ^2 and e**

Model-Predicted Item Covariance Matrix

- Matrix equation: $\boldsymbol{\Sigma} = \boldsymbol{\Lambda\Phi\Lambda}^T + \boldsymbol{\Psi}$

Σ= model-predicted item covariance matrix is created from:

Λ = item factor loadings

Φ= factor variances and covariances

Λ^T= item factor loadings transposed (~λ^2)

Ψ = item error variances

$$
\begin{pmatrix}
\sigma^2_{y1} & \sigma_{y1,y2} & \sigma_{y1,y3} & \sigma_{y1,y4} & \sigma_{y1,y5} & \sigma_{y1,y6} \\
\sigma_{y2,y1} & \sigma^2_{y2} & \sigma_{y2,y3} & \sigma_{y2,y4} & \sigma_{y2,y5} & \sigma_{y2,y6} \\
\sigma_{y3,y1} & \sigma_{y3,y2} & \sigma^2_{y3} & \sigma_{y3,y4} & \sigma_{y3,y5} & \sigma_{y3,y6} \\
\sigma_{y4,y1} & \sigma_{y4,y2} & \sigma_{y4,y3} & \sigma^2_{y4} & \sigma_{y4,y5} & \sigma_{y4,y6} \\
\sigma_{y5,y1} & \sigma_{y5,y2} & \sigma_{y5,y3} & \sigma_{y5,y4} & \sigma^2_{y5} & \sigma_{y5,y6} \\
\sigma_{y6,y1} & \sigma_{y6,y2} & \sigma_{y6,y3} & \sigma_{y6,y4} & \sigma_{y6,y5} & \sigma^2_{y6}
\end{pmatrix}
$$

$$
= \begin{pmatrix}
\lambda_{11} & 0 \\
\lambda_{21} & 0 \\
\lambda_{31} & 0 \\
0 & \lambda_{42} \\
0 & \lambda_{52} \\
0 & \lambda_{62}
\end{pmatrix}
\begin{pmatrix}
\sigma^2_{F1} & \sigma_{F1,F2} \\
\sigma_{F2,F1} & \sigma^2_{F2}
\end{pmatrix}
\begin{pmatrix}
\lambda_{11} & \lambda_{21} & \lambda_{31} & 0 & 0 & 0 \\
0 & 0 & 0 & \lambda_{42} & \lambda_{52} & \lambda_{62}
\end{pmatrix}
$$

$$
+ \begin{pmatrix}
\sigma^2_{e1} & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma^2_{e2} & 0 & 0 & 0 & 0 \\
0 & 0 & \sigma^2_{e3} & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma^2_{e4} & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma^2_{e5} & 0 \\
0 & 0 & 0 & 0 & 0 & \sigma^2_{e6}
\end{pmatrix}
$$

Model-Predicted Item Covariance Matrix

- $\boldsymbol{\Sigma} = \boldsymbol{\Lambda\Phi\Lambda}^{T} + \boldsymbol{\Psi}$ → **Predicted Covariance Matrix**

The **loadings** control how related items from the same factor are predicted to be.

Items within Factor 1

$$\begin{pmatrix}
\lambda_{11}^2\sigma_{F1}^2+\sigma_{e1}^2 & \lambda_{11}\sigma_{F1}^2\lambda_{21} & \lambda_{11}\sigma_{F1}^2\lambda_{31} & \lambda_{11}\sigma_{F2,F1}\lambda_{42} & \lambda_{11}\sigma_{F2,F1}\lambda_{52} & \lambda_{11}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{21}\sigma_{F1}^2\lambda_{11} & \lambda_{21}^2\sigma_{F1}^2+\sigma_{e2}^2 & \lambda_{21}\sigma_{F1}^2\lambda_{31} & \lambda_{21}\sigma_{F2,F1}\lambda_{42} & \lambda_{21}\sigma_{F2,F1}\lambda_{52} & \lambda_{21}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{31}\sigma_{F1}^2\lambda_{11} & \lambda_{31}\sigma_{F1}^2\lambda_{21} & \lambda_{31}^2\sigma_{F1}^2+\sigma_{e2}^2 & \lambda_{31}\sigma_{F2,F1}\lambda_{42} & \lambda_{31}\sigma_{F2,F1}\lambda_{52} & \lambda_{31}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{42}\sigma_{F2,F1}\lambda_{11} & \lambda_{42}\sigma_{F2,F1}\lambda_{21} & \lambda_{42}\sigma_{F2,F1}\lambda_{31} & \lambda_{42}^2\sigma_{F2}^2+\sigma_{e4}^2 & \lambda_{42}\sigma_{F2}^2\lambda_{52} & \lambda_{42}\sigma_{F2}^2\lambda_{62} \\
\lambda_{52}\sigma_{F2,F1}\lambda_{11} & \lambda_{52}\sigma_{F2,F1}\lambda_{21} & \lambda_{52}\sigma_{F2,F1}\lambda_{31} & \lambda_{52}\sigma_{F2}^2\lambda_{42} & \lambda_{52}^2\sigma_{F2}^2+\sigma_{e5}^2 & \lambda_{52}\sigma_{F2}^2\lambda_{62} \\
\lambda_{62}\sigma_{F2,F1}\lambda_{11} & \lambda_{62}\sigma_{F2,F1}\lambda_{21} & \lambda_{62}\boxed{\sigma_{F2,F1}}\lambda_{31} & \lambda_{62}\sigma_{F2}^2\lambda_{42} & \lambda_{62}\sigma_{F2}^2\lambda_{52} & \lambda_{62}^2\sigma_{F2}^2+\sigma_{e6}^2
\end{pmatrix}$$

Items within Factor 2

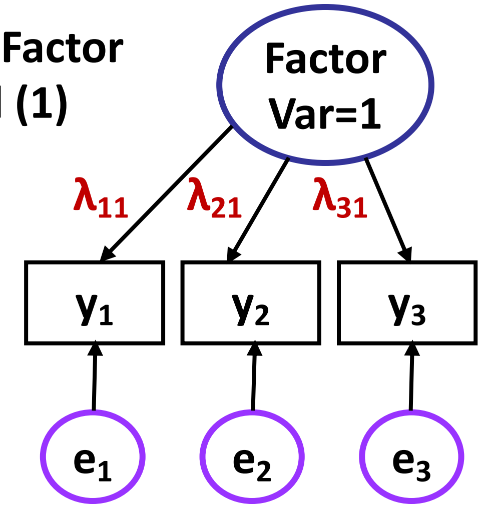
The only reason why items from different factors should be related is the **covariance** between the two factors.

The **loadings** also control how much of the item response is due to factor versus error.

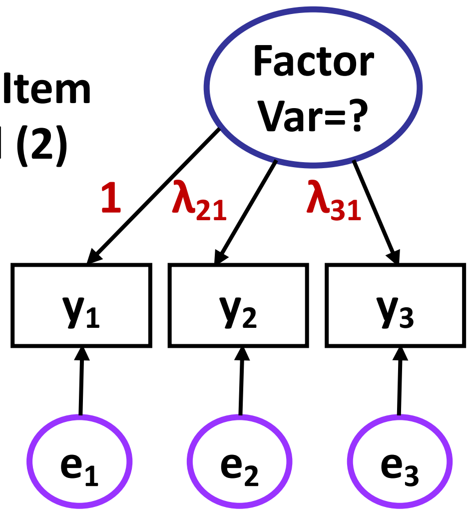
CFA Model Identification:

Create a Scale for the Latent Variable __Variance__

Z-Score Factor Method (1)



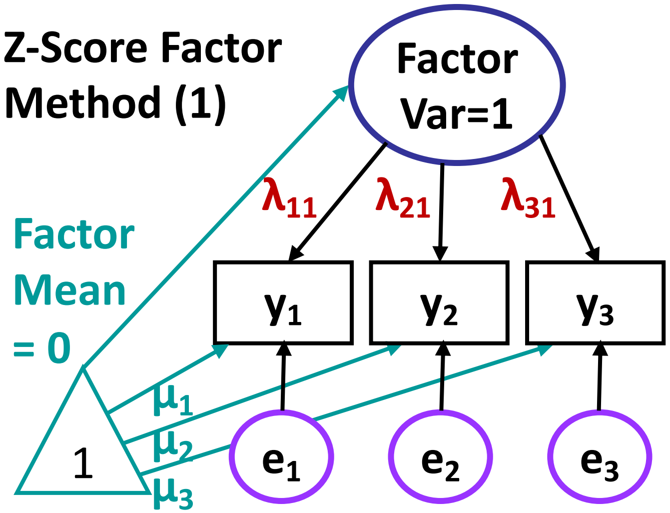
Marker Item Method (2)

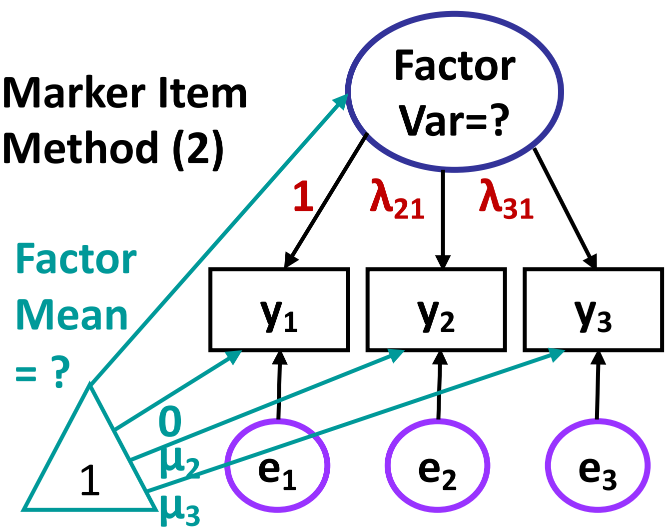


- The factor doesn't exist, so **it needs a scale** (it needs a mean and variance):
- Two **equivalent** options to create a scale for the factor **VARIANCE**:
  - **(1) Fix factor variance to 1: "z-score"**
    - Factor is interpreted as standardized
    - **__Can't__** be used in models with higher-order factors (coming later in the course)
  - **(2) Fix a "marker item" loading to 1**
    - Factor variance is then estimated the "reliable" part of the marker item variance
    - Std. loading = 0.9, item variance = 16? Factor variance = $(0.9^2)*16 = 12.96$
    - Can cause the model to blow up if marker item has no correlation with the factor at all

CFA Model Identification:

Create a Scale for the Latent Variable <u>Mean</u>



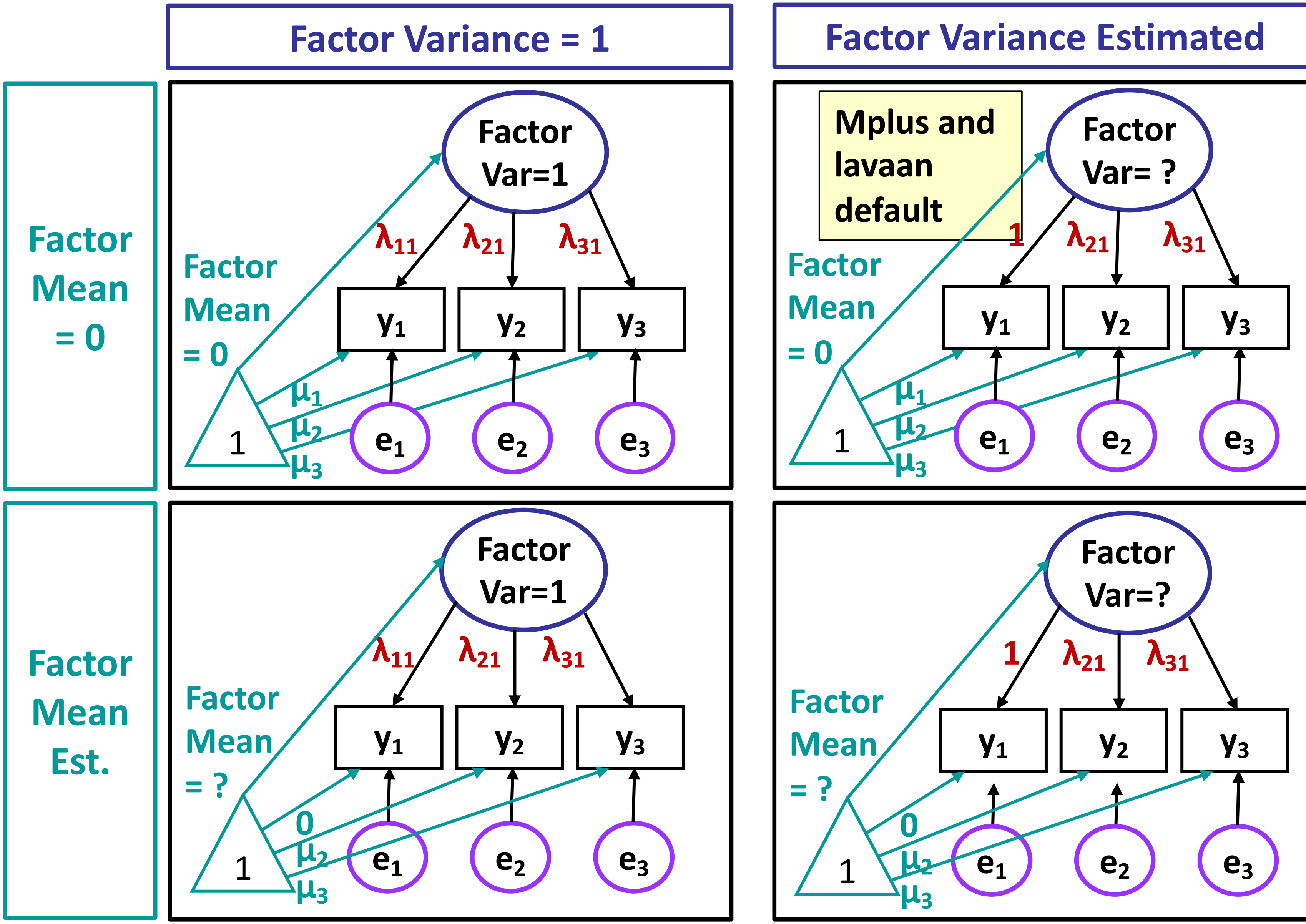


- The factor doesn't exist, so **it needs a scale** (it needs a mean and variance):
- Two **equivalent** options to create a scale for the factor **MEAN**:

  - **(1) Fix factor mean to 0: "z-score"**
    - Factor is interpreted as standardized
    - <u>**Can**</u> be used in models with higher-order factors (coming later in the course)
    - Item intercepts = item means

  - **(2) Fix a "marker item" intercept to 0**
    - Factor mean = mean of marker item
    - Item intercepts = expected item responses when factor = 0 (→ marker = 0)

Equivalent CFA Model Identifications



Factor Model Identification

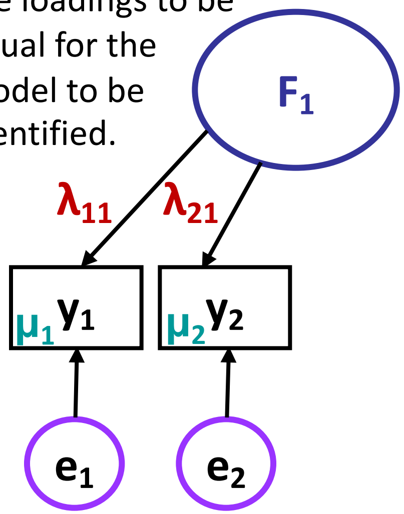
- Goal: *Name that Tune* → Reproduce observed item covariance matrix using as few estimated parameters as possible
  - (Robust) Maximum likelihood used to estimate model parameters
    - **Measurement Model**: Item factor loadings, item intercepts, item error variances
    - **Structural Model**: Factor variances and covariances, factor means
  - Global model fit is evaluated as difference between model-predicted matrix and observed matrix (but only the covariances really contribute to misfit)

- How many possible parameters can you estimate (total DF)?
  - **Total DF depends on # ITEMS → $\mathbf{v}$** (NOT on # people)
  - Total number of **unique elements** in item covariance matrix
    - Unique item elements = each variance, each covariance, each mean
    - Total unique elements = $(\mathrm{v}(\mathrm{v}+1)\ /\ 2)+\mathrm{v}$ → if 4 items, then ((4*5)/2) + 4 = 14
- Model degrees of freedom (df) = data input − model output
  - Model df = # possible parameters − # estimated parameters

Under-Identified Factor: 2 Items

- Model is <u>under-identified</u> if there are more unknown parameters then item variances, covariances, and means with which to estimate them
  - Cannot be solved because there are an infinite number of different parameter estimates that would result in perfect fit
  - Example: x + y = 7 ??

You'd have to constrain the loadings to be equal for the model to be identified.



In other words, the assumptions required to calculate reliability in CTT are the result of model under-identification.

Total possible df = unique pieces of data = **5**

0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
2 item loadings	**OR**	**1 item loading**
2 item intercepts		**1 item intercept**
2 error variances		**2 error variances**

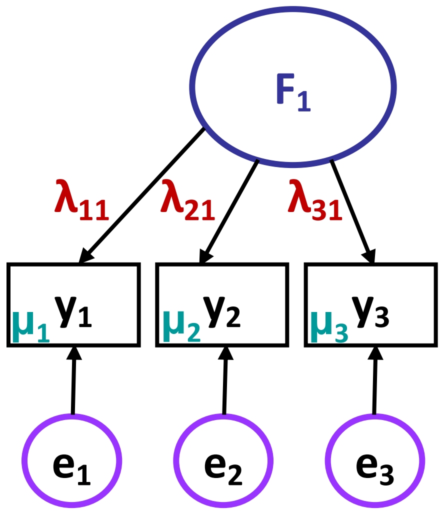
df = 5 – 6 = – 1

If $r_{y1,y2} = .64$, then:

$\lambda_{11} = .800$, $\lambda_{21} = .800$??
$\lambda_{11} = .900$, $\lambda_{21} = .711$??
$\lambda_{11} = .750$, $\lambda_{21} = .853$??

Just-Identified Factor: 3 Items

- Model is <u>just-identified</u> if there are as many unknown parameters as item variances, covariances, and means with which to estimate them
  - Parameter estimates have a unique solution that will perfectly reproduce the observed matrix
  - Example: Solve x + y = 7, 3x – y = 1



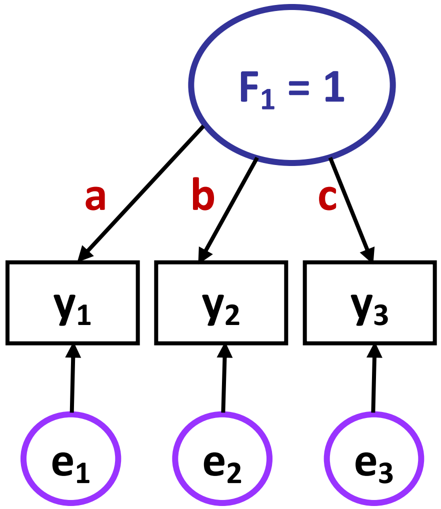
Total possible **df** = unique pieces of data = **9**

0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
3 item loadings	**OR**	**2 item loadings**
3 item intercepts		**2 item intercepts**
3 error variances		**3 error variances**

df = 9 – 9 = 0

Not really a model—more like a description

Solving a Just-Identified Model

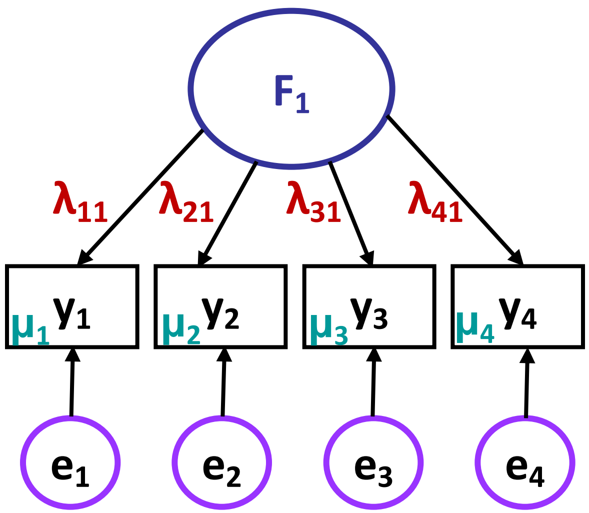


	Y_1	Y_2	Y_3
Y_1	1.00		
Y_2	.595	1.00	
Y_3	.448	.544	1.00

- Step 1: $ab = .595$
  $ac = .448$
  $bc = .544$
- Step 2: $b = .595/a$
  $c = .488/a$
  $(.595/a)(.448/a) = .544$
- Step 3: $.26656/a^2 = .544$
  $a = .70$
- Step 4: $.70b = .595 \quad b = .85$
  $.70c = .448 \quad c = .64$
- Step 5: $Var(e_1) = 1 - a^2 = .51$

Over-Identified Factor: 4+ Items

- Model is <u>over-identified</u> if there are fewer unknown parameters than item variances, covariances, and means with which to estimate them
  - Parameter estimates have a unique solution that will NOT perfectly reproduce the observed matrix → **if df > 0, NOW we can test model fit**



Total possible **df** = unique pieces of data = **14**

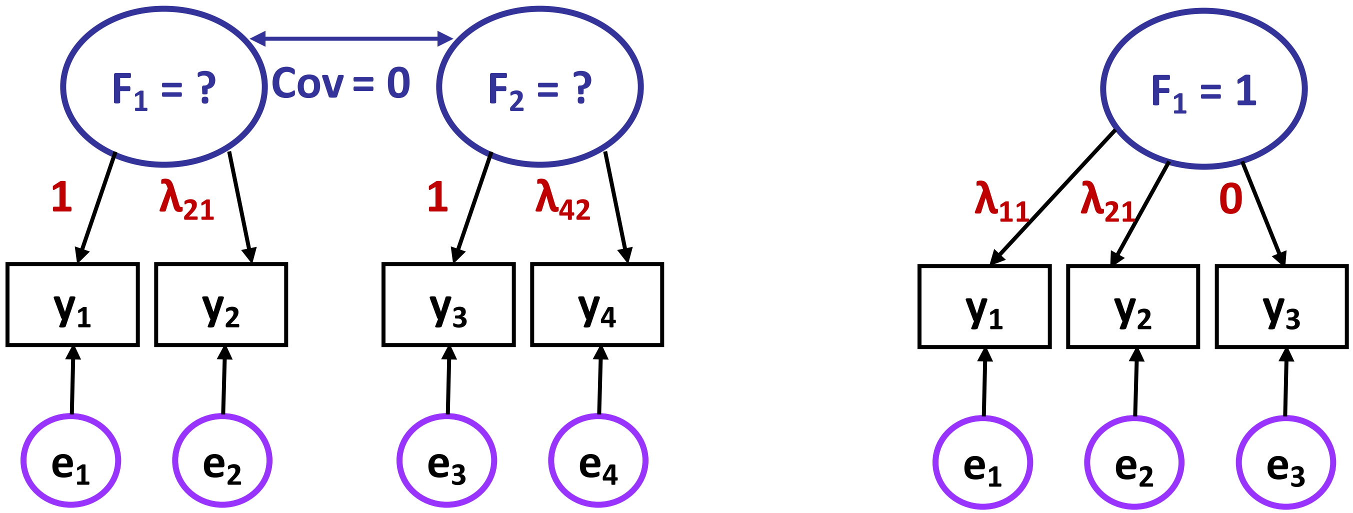
0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
4 item loadings	**OR**	**3 item loadings**
4 item intercepts		**3 item intercepts**
4 error variances		**4 error variances**

df = 14 – 12 = 2

Did we do a 'good enough' job reproducing the item covariance matrix with 2 fewer parameters than it was possible to use?

Oops: Empirical Under-Identification

- Did your model blow up (errors instead of output)?
  - Make sure each factor is identified (scale of factor mean and variance is set)
- Sometimes you can set up your model correctly and it will still blow up because of **empirical under-identification**
  - It's not you; **it's your data** – here are two examples of when these models should have been identified, but weren't because of an unexpected 0 relationship



Intermediate Summary: CFA

- CFA is a **linear model** in which continuous observed item responses are predicted from latent factors (traits) and error
  - Goal is to reproduce observed **item covariance matrix** using estimated parameters (intercept, loading, and error variance for items, factor variance)
  - Factor model makes specific testable mathematical predictions about how item responses should relate to each other: **loadings predict covariances**
  - Need at least 3 items per factor for the model to be identified; need **at least 4 items for model fit to be testable**

- CFA framework offers significant advantages over CTT by offering the potential for comparability across samples, groups, and time
  - CTT: No separation of observed item responses from true score
    - Sum across items = true score; item properties belong to that sample only
  - CFA: Latent trait is estimated separately from item responses
    - Separates interpretation of person traits from specific items given
    - Separates interpretation of item properties from specific persons in sample

The Big Picture of Model Fit

- Aspects of the observed data to be predicted *(assuming a z-score metric for the factor for simplicity)*:
- CFA model equation: $\mathbf{y_{is}} = \boldsymbol{\mu}_\mathbf{i} + \boldsymbol{\lambda}_\mathbf{i}\mathbf{F_s} + \mathbf{e_{is}}$
  - **Mean** per item: Predicted by intercept $\boldsymbol{\mu}_\mathbf{i}$ per item
    - Not a source of misfit (unless constraints are applied on the intercepts)
  - **Variance** per item: Predicted by weighted $(\mathbf{F})$ + $(\mathbf{e})$
    - $\mathbf{Var(y_i)} = \boldsymbol{\lambda}_\mathbf{i}^\mathbf{2} * \mathbf{Var(F)} + \mathbf{Var(e_i)}$ → output is given as $\boldsymbol{\lambda}_\mathbf{i}$ and $\mathbf{Var(e_i)}$ → $\mathbf{e_i^2}$
    - Factor and error variances are additive → not a source of misfit (whatever F doesn't get, e_i picks up to get back to total y_i variance)
  - **Covariance** among items: Predicted via factor loadings $\boldsymbol{\lambda}_\mathbf{i}$
    - Loadings multiplied predict what observed covariance should be… but they may not be right → **THE PRIMARY SOURCE OF MISFIT**

Baselines for Assessing Fit in CFA (Item means all saturated in both)

Independence (Null) Model

$$\begin{pmatrix} \sigma^2_{y1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{y2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{y3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{y4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{y5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{y6} \end{pmatrix}$$

Saturated (Unstructured; H1) Model

$$\begin{pmatrix} \sigma^2_{y1} & \sigma_{y1,y2} & \sigma_{y1,y3} & \sigma_{y1,y4} & \sigma_{y1,y5} & \sigma_{y1,y6} \\ \sigma_{y2,y1} & \sigma^2_{y2} & \sigma_{y2,y3} & \sigma_{y2,y4} & \sigma_{y2,y5} & \sigma_{y2,y6} \\ \sigma_{y3,y1} & \sigma_{y3,y2} & \sigma^2_{y3} & \sigma_{y3,y4} & \sigma_{y3,y5} & \sigma_{y3,y6} \\ \sigma_{y4,y1} & \sigma_{y4,y2} & \sigma_{y4,y3} & \sigma^2_{y4} & \sigma_{y4,y5} & \sigma_{y4,y6} \\ \sigma_{y5,y1} & \sigma_{y5,y2} & \sigma_{y5,y3} & \sigma_{y5,y4} & \sigma^2_{y5} & \sigma_{y5,y6} \\ \sigma_{y6,y1} & \sigma_{y6,y2} & \sigma_{y6,y3} & \sigma_{y6,y4} & \sigma_{y6,y5} & \sigma^2_{y6} \end{pmatrix}$$

Parsimony ⟷ **Good fit**

All item means and variances estimated separately; no covariances are estimated

$$y_{is} = \mu_i + \cancel{\lambda_i F_s} + e_{is}$$

All item means and variances estimated separately; all covariances estimated are separately now, too.

Your CFA "H0" model will go here

4 Steps in Assessing Model Fit

1. Global model fit
   - *Does the model 'work' as a whole?*
2. Local model fit
   - *Are there any more specific problems?*
3. Inspection of model parameters
   - *Are the estimates, SEs, and the item responses they predict plausible?*
4. Reliability and information per item
   - *How 'good' is my test? How useful is each item?*

Posterior Predictive Model Checks

- To evaluate a model using Bayesian methods, we use posterior predictive model checks

- A posterior predictive model check is conducted following sampling from the posterior and ensuring the Markov chains have converged

- Steps in a PPMC:
  - 1. Sample one iteration of the Markov chain from the posterior samples section
  - 2. Generate a new data set where the values of the parameters of the model distribution are the values from the iteration in #1
  - 3. Calculate selected summary statistics from the simulated data
  - 4. Repeat 1-3 many times
  - 5. Compare the final result with values from the data
    - Note: this is where the Bayesians all get ML-happy as “values from the data” are typically MLEs

Indices of Global Model Fit

- PPMC using typical CFA model fit statistics
- Absolute Fit: $\boldsymbol{\chi}^2$
  - Don't use "ratio rules" like $\chi^2/\mathrm{df} > 2$ or $\chi^2/\mathrm{df} > 3$
- Absolute Fit: **SRMR**
  - **Standardized Root Mean Square Residual**
  - Get difference of standardized $\boldsymbol{\Sigma}$ and $\boldsymbol{S}$ → residual matrix
  - Sum the squared residuals of the correlation matrix across items, divide by number of residuals (i.e., matrix elements)
  - Ranges from 0 to 1: smaller is better
  - ".08 or less" → good fit
- See also: **RMR** (**Root Mean Square Residual**)

Indices of Global Model Fit

Parsimony-Corrected: **RMSEA**

- **Root Mean Square Error of Approximation**
- Relies on a "non-centrality parameter" (NCP)
  - Indexes how far off your model is → χ^2 distribution shoved over
  - NCP → $d = (\chi^2 - df) / N$ Then, $RMSEA = \sqrt{d/df}$
- RMSEA ranges from 0 to 1; smaller is better
  - < .05 or .06 = "good", .05 to .08 = "acceptable", .08 to .10 = "mediocre", and >.10 = "unacceptable"
  - In addition to point estimate, get 90% confidence interval
  - RMSEA penalizes for model complexity – it's discrepancy in fit per df left in model (but not sensitive to N, although CI can be)
  - Test of "close fit": null hypothesis that RMSEA ≤ .05

Indices of Global Model Fit

Comparative (Incremental) Fit Indices

- Fit evaluated relative to a 'null' or 'independence' model (of 0 covariances)
- Relative to that, your model fit should be great!

- **CFI: Comparative Fit Index**
  - Also based on idea of NCP (χ^2 – df)
  - $$\text{CFI} = 1 - \frac{\max[(\chi^2_T - df_T), 0]}{\max[(\chi^2_T - df_T), (\chi^2_N - df_N), 0]}$$
    T = target model
    N = null model
  - From 0 to 1: bigger is better, > .90 = "acceptable", > .95 = "good"

- **TLI: Tucker-Lewis Index** (= **Non-Normed Fit Index**)
  - $$\text{TLI} = \frac{(\chi^2_N/df_N) - (\chi^2_T/df_T)}{(\chi^2_N/df_N) - 1}$$
  - From <0 to >1, bigger is better, >.95 = "good"

Summary: Steps 1 and 2

1. **Assess global model fit**
   - Recall that item intercepts, factor means, and variances are usually just-identified → *so misfit comes from messed-up covariances*
   - χ^2 is sensitive to large sample size, so pick at least one global fit index from each class (e.g., CFI, RMSEA); hope they agree
2. **Identify localized model strain**
   - Global model fit means that the observed and predicted covariance matrices aren't too far off on the whole… says nothing about the specific matrix elements (reproduction of each covariance)
   - Consider normalized residuals and modification indices to try and "fix" the model (add or remove factors, add or remove error covariances, etc.) – Has to be theoretically justifiable!!

Good global and local fit? Great, but we're not done yet…